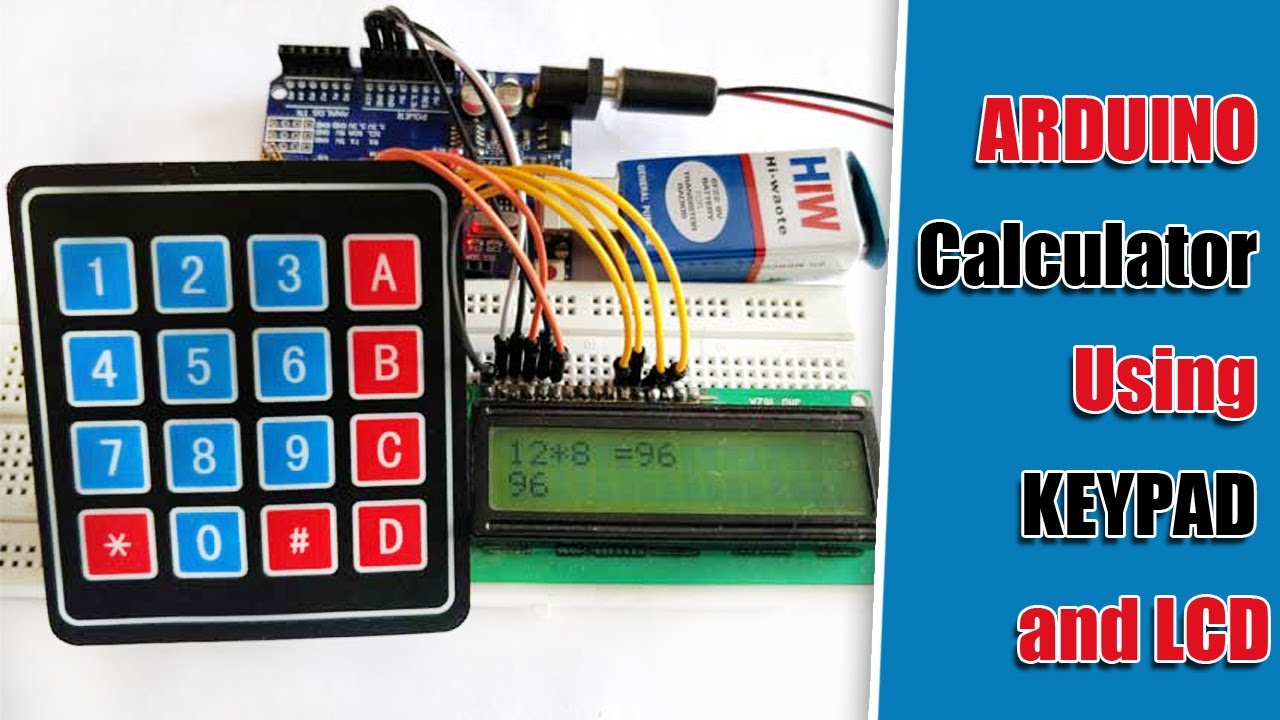
CALCULATOR ARDUINO



FACULTATEA: AUTOMATICA SI CALCULATOARE

NUME: DOMIDE MARIA

GRUPA 30121

**PRINCIPALA IDEE**

Principala idee a proiectului constă în dezvoltarea unui calculator auditiv, destinat utilizării persoanelor cu deficiențe de vedere. Acesta folosește sunete specifice pentru a reprezenta cifrele, operațiile matematice și rezultatele, permițând utilizatorului să interacționeze și să efectueze operații matematice fără a avea nevoie de informații vizuale.

**SCOPUL PROIECTULUI**

Scopul acestui proiect constă în furnizarea unei interfețe accesibile și intuitive pentru persoanele nevăzătoare, si nu numai,permițându-le să efectueze calcule matematice fundamentale. Prin utilizarea sunetelor distincte pentru cifre și operații, acest calculator auditiv vizează să ofere o soluție simplă și eficientă pentru realizarea adunărilor, scăderilor, înmulțirilor și împărțirilor. Această interfață sonoră își propune să ofere o modalitate simplificată și intuitivă de interacțiune, eliminând barierele legate de vizual și facilitând procesul de învățare și utilizare a funcțiilor de bază matematice.

**AVANTAJELE PROIECTULUI**

Calculatorul auditiv pentru nevăzători oferă o modalitate inovatoare de a interacționa cu operațiile matematice de baza dar in acelasi timp si pentru copii sau noi cei care suntem dispusi sa invatam si sa ne dezvoltam.

**Accesibilitate și independență**: Prin utilizarea unui calculator auditiv, persoanele nevăzătoare pot efectua calcule matematice de bază fără a depinde de ecrane vizuale sau asistență constantă.

**Simplicitate și ușurință în utilizare:** Interfața auditivă elimină complexitatea interfeței vizuale, oferind o modalitate intuitivă și ușor de folosit pentru efectuarea calculelor.

**Confort:** Calculatorul auditiv poate reprezenta o alternativă mai confortabilă pentru utilizatorii nevăzători, evitând oboseala sau tensiunea asociată cu ecranele sau dispozitivele vizuale.

**Educație și învățare:** Poate servi ca un instrument educativ util pentru înțelegerea și practicarea matematicii de bază, oferind o modalitate interactivă de a explora operațiile matematice esențiale.

Avantajele acestui proiect nu se opresc doar la utilizarea de către persoanele nevăzătoare, ci se extind și către utilizatorii obișnuiți sau către copiii care învață să facă calcule.

Acest calculator auditiv oferă o modalitate interactivă și intuitivă de a învăța și practica operațiile matematice de bază. Sunetele distincte asociate cifrelor și operațiilor îl fac potrivit pentru utilizare în scop educațional, ajutându-i pe copii să asocieze sunetele cu numerele și operațiile matematice într-un mod ludic și captivant.

Astfel, acest proiect nu este doar un instrument util pentru persoanele cu nevoi speciale, ci și o resursă educativă interesantă și utilă pentru o gamă largă de utilizatori.

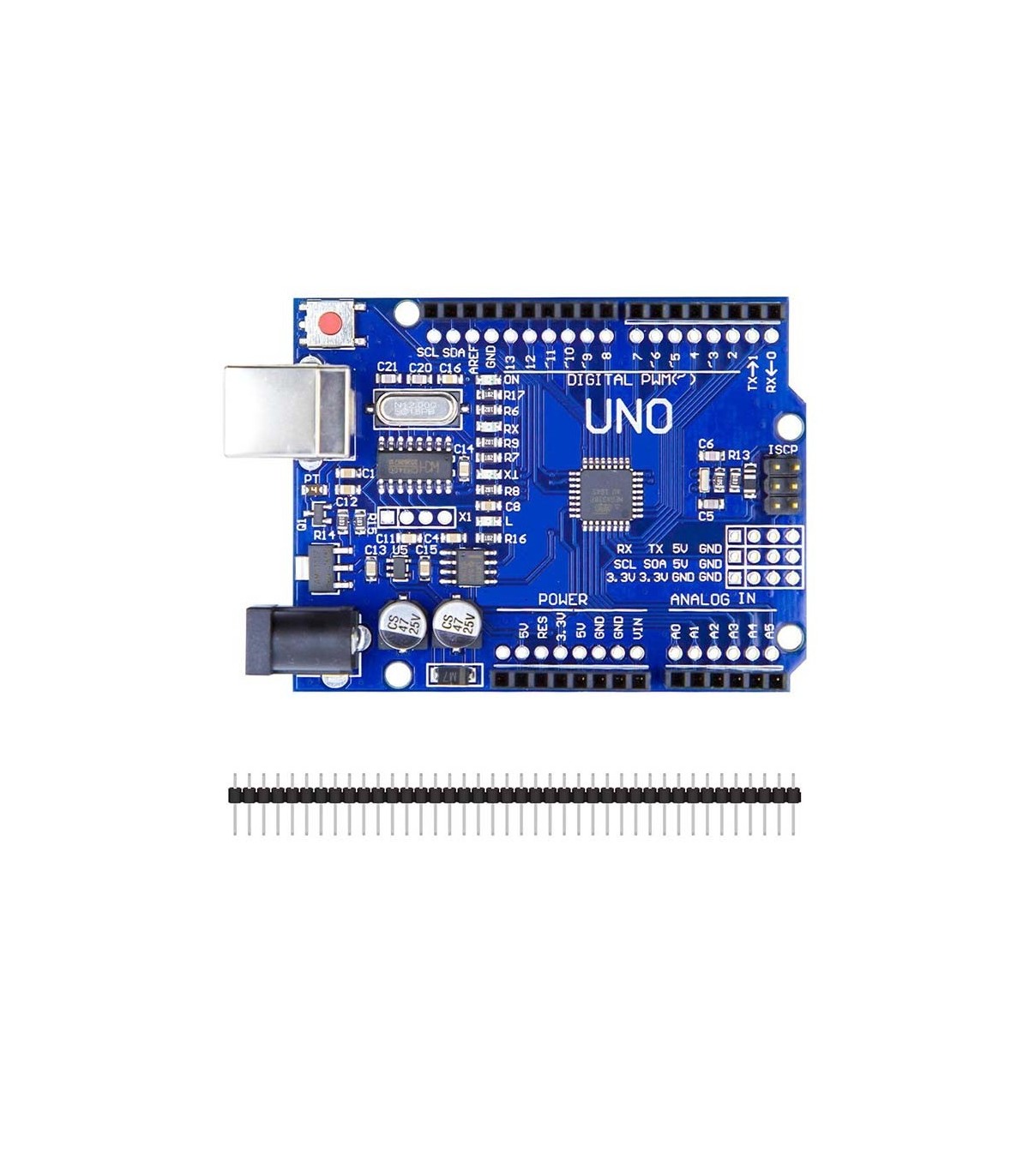
**DESCRIEREA TEHNICA**

Descrierea tehnică a proiectului se bazează pe utilizarea unei plăci Arduino UNO pentru controlul unui LCD și al unei tastaturi matricele, creand astfel un calculator matematic interactiv. Prin simpla apăsare a tastelor de pe tastatura matricială, se generează sunete distincte cu frecvențe specifice, reprezentând cifra sau operația matematică asociată. Rezultatul operației este de asemenea redat auditiv după apăsarea tastei de egal. Sunetele sunt generate prin intermediul unui buzzer conectat la placa Arduino, iar interacțiunile utilizatorului sunt afișate și transmise sonor pentru fiecare acțiune matematică realizată.

Proiectul include un buton de ON și OFF care controlează starea unui LED specific. În momentul în care butonul este apăsat și calculatorul este pornit, pe ecranul LCD se afișează un mesaj care indică faptul că calculatorul este activ și poate efectua operații matematice, în același timp generându-se un sunet specific. LED-ul indică starea ON. La apăsarea repetată a butonului, calculatorul se va opri, generând un alt sunet specific și afișând un mesaj pe ecranul LCD care indică faptul că acesta este oprit, în timp ce LED-ul indică starea OFF a calculatorului. Aceste elemente asigură o interacțiune auditivă și vizuală pentru utilizatorul nevăzător, facilitând procesul de utilizare a funcțiilor de bază matematice.

**COMPONENTE:**

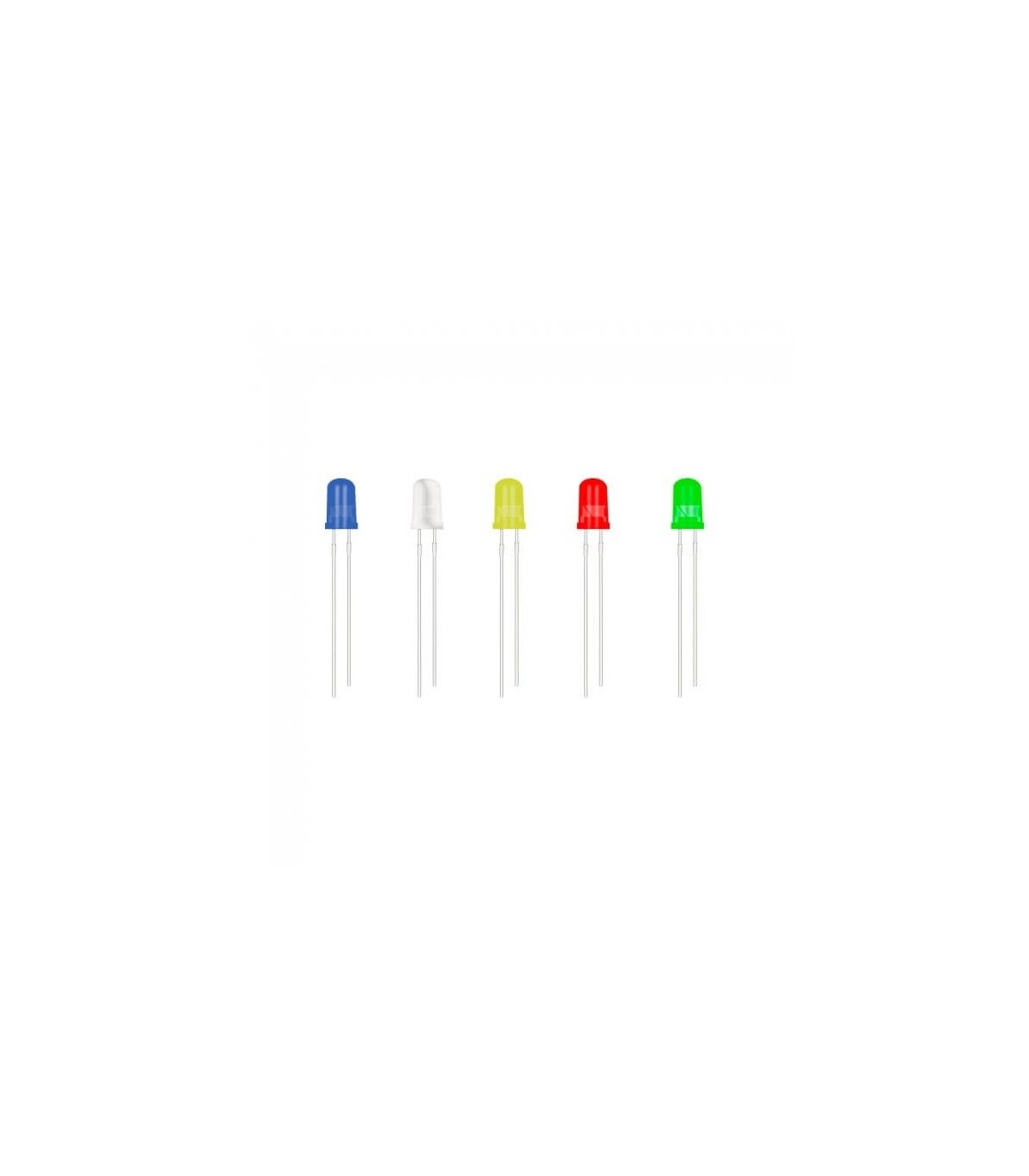
* Placa ARDUINO UNO



* BUTON



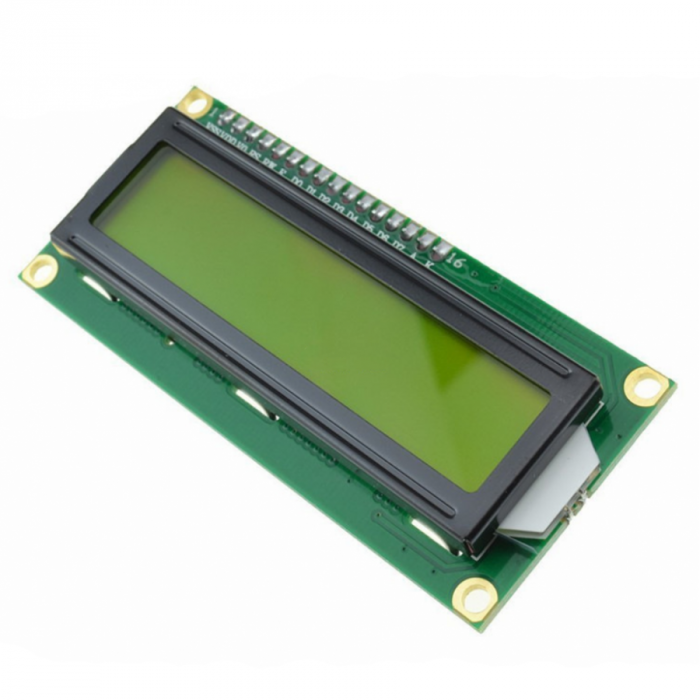
* LED-URI



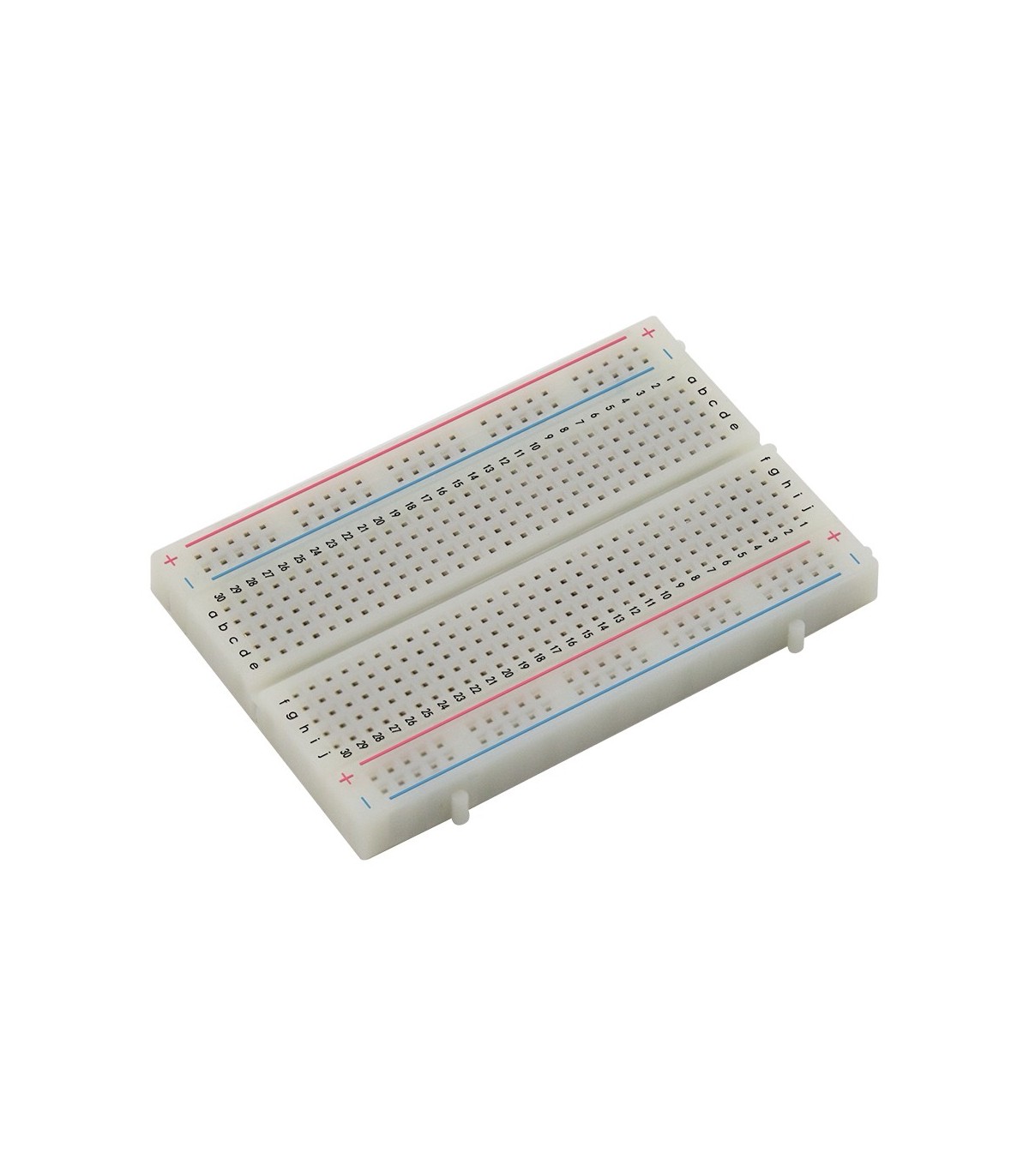
* TASTATURA NUMERICA 4X4



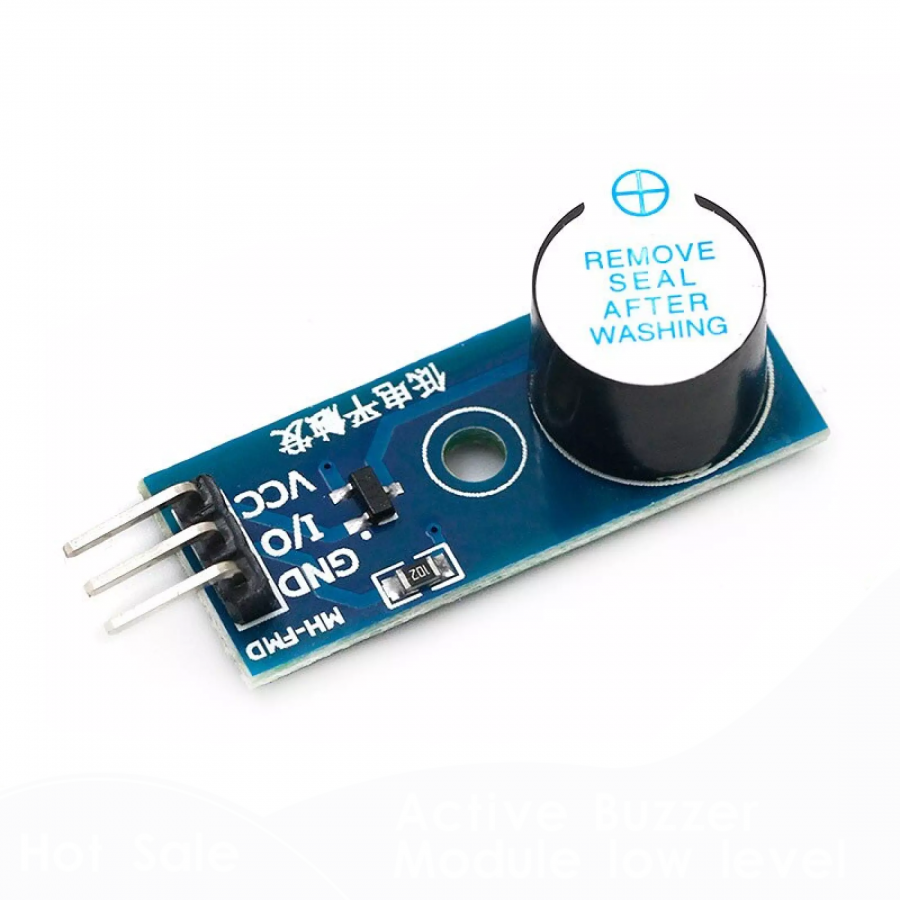
* ECRAN LCD



* BREADBOARD



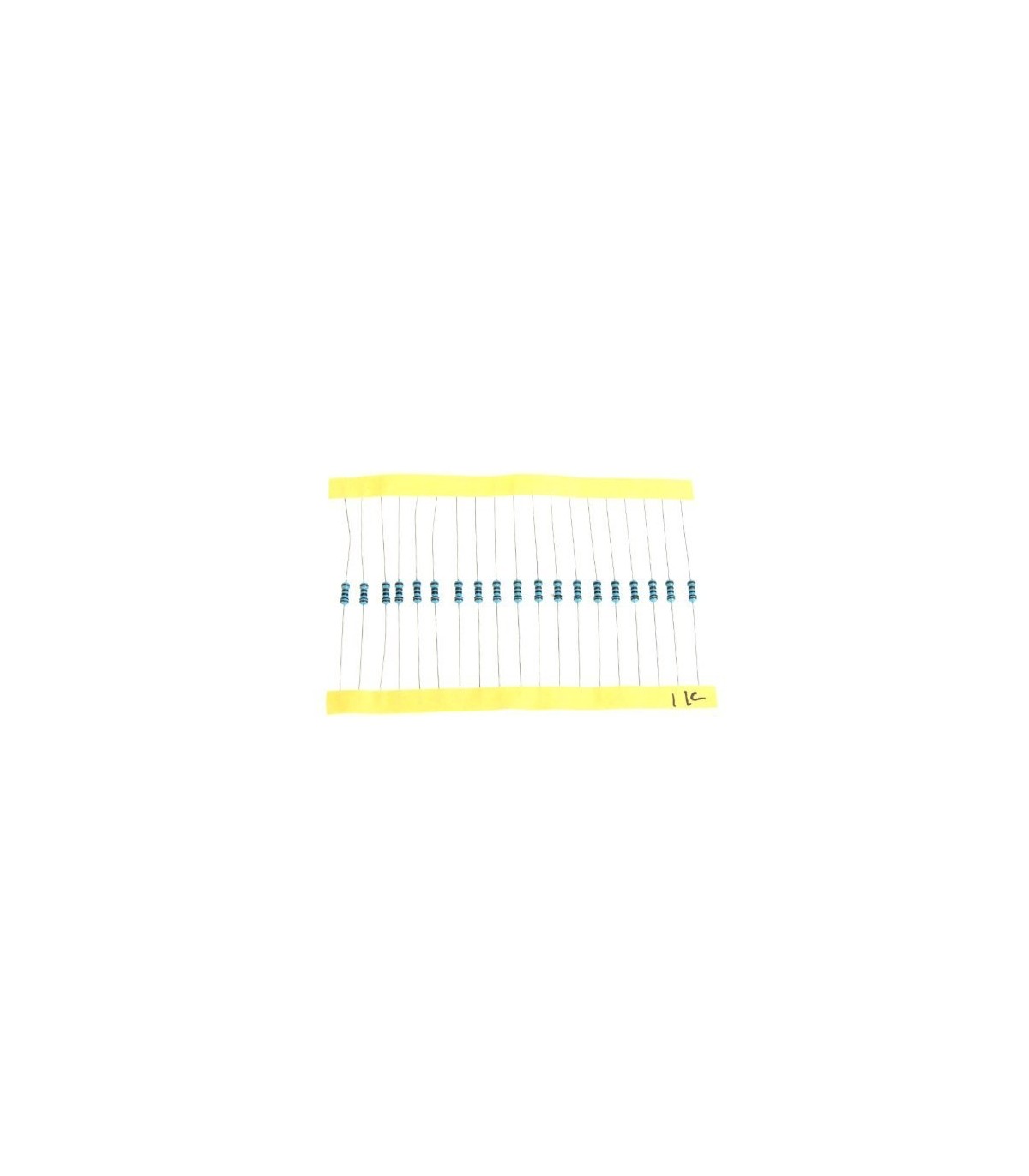
* BUZZER



* FIRE JUMPER /DUPONT



* REZISTENTE



* SURSA DE ALIMENTARE (BATERIE)



* CONECTOR BATERIE



**COD:**

#include <Wire.h> // Include biblioteca pentru comunicare I2C

#include <LiquidCrystal\_I2C.h> // Include biblioteca pentru LCD-ul I2C

#include <Keypad.h> // Include biblioteca pentru tastatură

int buzzer\_Pin = 10; // Pinul la care este conectat buzzer-ul

int led\_Pin = 11; // Pinul la care este conectat LED-ul

int button\_Pin = 8; // Pinul la care este conectat butonul

bool calculatorOn = false; // Variabila pentru a urmări starea calculatorului (pornit/oprit)

bool result = false; // Variabila pentru a urmări dacă este prezent un rezultat

int Number = 0; // Variabila pentru a stoca un număr introdus

int Num1 = 0; // Variabila pentru primul număr din operație

int Num2 = 0; // Variabila pentru al doilea număr din operație

char action = ' '; // Variabila pentru a stoca operația matematică

int sound\_On = 3000; // Sunet pentru starea "ON"

int sound\_Off = 2000; // Sunet pentru starea "OFF"

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Inițializare LCD cu adresa 0x27 și dimensiunile 16x2

const byte ROWS = 4; // Numărul de rânduri ale tastaturii

const byte COLS = 4; // Numărul de coloane ale tastaturii

char keys[ROWS][COLS] = { // Matricea pentru harta tastelor

    {'1', '2', '3', 'A'},

    {'4', '5', '6', 'B'},

    {'7', '8', '9', 'C'},

    {'\*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {0, 1, 2, 3}; // Vectorul pentru pini rând

byte colPins[COLS] = {4, 5, 6, 7}; // Vectorul pentru pini coloană

Keypad kpd = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS); // Inițializare tastatură

void setup() {

    pinMode(led\_Pin, OUTPUT); // Setează pinul LED-ului ca ieșire

    pinMode(button\_Pin, INPUT); // Setează pinul butonului ca intrare

    lcd.begin(); // Inițializează LCD-ul

    lcd.backlight(); // Activează iluminarea LCD-ului

    lcd.print("DIY Calculator"); // Afișează un mesaj de pornire pe LCD

    lcd.setCursor(0, 1);

    lcd.print("-CircuitDigest");

    delay(2000);

    lcd.clear();

    digitalWrite(led\_Pin, LOW); // Oprește LED-ul la pornire

}

void loop() {

    // Variabila statică pentru a reține starea butonului

    static bool wasPressed = false;

    // Verifică dacă butonul este apăsat și nu a fost apăsat anterior

    if (digitalRead(button\_Pin) == LOW && !wasPressed) {

        // Setează wasPressed ca fiind true pentru a indica că butonul a fost apăsat

        wasPressed = true;

        // Verifică și gestionează starea calculatorului

        if (!calculatorOn) {

            // Pornește calculatorul și afișează "Calculator ON" pe LCD

            calculatorOn = true;

            digitalWrite(led\_Pin, HIGH);

            lcd.clear();

            lcd.setCursor(0, 0);

            lcd.print("Calculator ON ");

            delay(500);

            // Generează sunet pentru starea "ON"

            tone(buzzer\_Pin, sound\_On, 200);

            delay(200);

            noTone(buzzer\_Pin);

        } else {

            // Oprește calculatorul și afișează "Calculator OFF" pe LCD

            calculatorOn = false;

            digitalWrite(led\_Pin, LOW);

            lcd.clear();

            lcd.setCursor(0, 0);

            lcd.print("Calculator OFF");

            delay(500);

            // Generează sunet pentru starea "OFF"

            tone(buzzer\_Pin, sound\_Off, 200);

            delay(200);

            noTone(buzzer\_Pin);

        }

    } else if (digitalRead(button\_Pin) == HIGH) {

        // Dacă butonul este eliberat, actualizează wasPressed pentru a permite detectarea ulterioară a apăsării butonului

        wasPressed = false;

    }

    // Verifică dacă calculatorul este pornit și gestionează comenzile de la tastatură

    if (calculatorOn) {

        char key = kpd.getKey();

        if (key != NO\_KEY) {

            DetectButtons(key); // Apel la funcția care gestionează butoanele

            DisplayResult(); // Afișează rezultatul

            if (result == true) {

                CalculateResult(); // Calculează rezultatul

                DisplayResult(); // Afișează rezultatul calculat

                Num1 = Num2 = Number = 0; // Resetarea variabilelor pentru următoarea operație

                result = false;

            }

        }

    }

}

void DetectButtons(char key) {

    lcd.clear(); // Curăță ecranul LCD

    if (key == '\*') {

        Serial.println("Button Cancel"); // Afișează în Serial Monitor mesajul pentru anularea operației

        Number = Num1 = Num2 = 0; // Resetează variabilele pentru a anula operația

        result = false; // Setează rezultatul ca fiind fals

    } else if (key >= '0' && key <= '9') {

        if (Number == 0) {

            Number = key - '0'; // Converteste caracterul tastat in numar

        } else {

            Number = (Number \* 10) + (key - '0'); // Construiește numărul din mai multe cifre

        }

        // Generează sunet specific pentru cifră

        switch (key) {

            case '0':

                tone(buzzer\_Pin, 470, 200);

                break;

            case '1':

                tone(buzzer\_Pin, 523, 200);

                break;

            case '2':

                tone(buzzer\_Pin, 700, 200);

                break;

            case '3':

                tone(buzzer\_Pin, 410, 200);

                break;

            case '4':

                tone(buzzer\_Pin, 998, 200);

                break;

            case '5':

                tone(buzzer\_Pin, 120, 200);

                break;

            case '6':

                tone(buzzer\_Pin, 250, 200);

                break;

            case '7':

                tone(buzzer\_Pin, 310, 200);

                break;

            case '8':

                tone(buzzer\_Pin, 380, 200);

                break;

            case '9':

                tone(buzzer\_Pin, 1030, 200);

                break;

        }

        delay(200); // O mică pauză între sunet și oprirea sunetului

        noTone(buzzer\_Pin); // Oprirea sunetului

    } else if (key == '#') {

        Serial.println("Button Equal"); // Afișează în Serial Monitor mesajul pentru egal

        Num2 = Number; // Salvează numărul introdus pentru a fi folosit în operație

        result = true; // Setează rezultatul ca fiind adevărat pentru a calcula operația

        // Generează sunet pentru egal

        tone(buzzer\_Pin, 950, 200);

        delay(200);

        noTone(buzzer\_Pin);

    } else if (key == 'A') {

        Serial.println("Addition"); // Afișează în Serial Monitor mesajul pentru adunare

        action = '+'; // Setează operația curentă ca fiind adunare

        Num1 = Number; // Salvează numărul introdus pentru a fi folosit în operație

        Number = 0; // Resetează numărul pentru introducerea celui de-al doilea număr

        // Generează sunet pentru adunare

        tone(buzzer\_Pin, 660, 200);

        delay(200);

        noTone(buzzer\_Pin);

    } else if (key == 'B') {

        Serial.println("Subtraction"); // Afișează în Serial Monitor mesajul pentru scădere

        action = '-'; // Setează operația curentă ca fiind scădere

        Num1 = Number; // Salvează numărul introdus pentru a fi folosit în operație

        Number = 0; // Resetează numărul pentru introducerea celui de-al doilea număr

        // Generează sunet pentru scădere

        tone(buzzer\_Pin, 880, 200);

        delay(200);

        noTone(buzzer\_Pin);

    } else if (key == 'C') {

        Serial.println("Multiplication"); // Afișează în Serial Monitor mesajul pentru înmulțire

        action = '\*'; // Setează operația curentă ca fiind înmulțire

        Num1 = Number; // Salvează numărul introdus pentru a fi folosit în operație

        Number = 0; // Resetează numărul pentru introducerea celui de-al doilea număr

        // Generează sunet pentru înmulțire

        tone(buzzer\_Pin, 910, 200);

        delay(200);

        noTone(buzzer\_Pin);

    } else if (key == 'D') {

        Serial.println("Division"); // Afișează în Serial Monitor mesajul pentru împărțire

        action = '/'; // Setează operația curentă ca fiind împărțire

        Num1 = Number; // Salvează numărul introdus pentru a fi folosit în operație

        Number = 0; // Resetează numărul pentru introducerea celui de-al doilea număr

        // Generează sunet pentru împărțire

        tone(buzzer\_Pin, 778, 200);

        delay(200);

        noTone(buzzer\_Pin);

    }

}

void CalculateResult() {

    if (action == '+')

        Number = Num1 + Num2; // Adunarea numerelor

    else if (action == '-')

        Number = Num1 - Num2; // Scăderea numerelor

    else if (action == '\*')

        Number = Num1 \* Num2; // Înmulțirea numerelor

    else if (action == '/')

        Number = Num1 / Num2; // Împărțirea numerelor

}

void DisplayResult() {

    lcd.setCursor(0, 0);

    lcd.print(Num1); // Afișează primul număr

    lcd.print(action); // Afișează simbolul operației

    lcd.print(Num2); // Afișează al doilea număr

    if (result == true) { // Verifică dacă rezultatul este disponibil pentru afișare

        lcd.print(" =");

        lcd.print(Number); // Afișează rezultatul

        // Parcurge fiecare cifră din rezultat și generează sunetul specific pentru fiecare cifră

        int temp = Number;

        while (temp != 0) {

            int digit = temp % 10; // Ia ultima cifră din rezultat

            switch (digit) {

                case 0:

                    tone(buzzer\_Pin, 470, 200);

                    break;

                case 1:

                    tone(buzzer\_Pin, 523, 200);

                    break;

                case 2:

                    tone(buzzer\_Pin, 700, 200);

                    break;

                case 3:

                    tone(buzzer\_Pin, 410, 200);

                    break;

                case 4:

                    tone(buzzer\_Pin, 938, 200);

                    break;

                case 5:

                    tone(buzzer\_Pin, 120, 200);

                    break;

                case 6:

                    tone(buzzer\_Pin, 250, 200);

                    break;

                case 7:

                    tone(buzzer\_Pin, 310, 200);

                    break;

                case 8:

                    tone(buzzer\_Pin, 380, 200);

                    break;

                case 9:

                    tone(buzzer\_Pin, 1030, 200);

                    break;

            }

            delay(200); // Pauză între sunetele cifrelor

            noTone(buzzer\_Pin); // Oprește sunetul pentru cifra respectivă

            temp /= 10; // Trece la următoarea cifră din rezultat

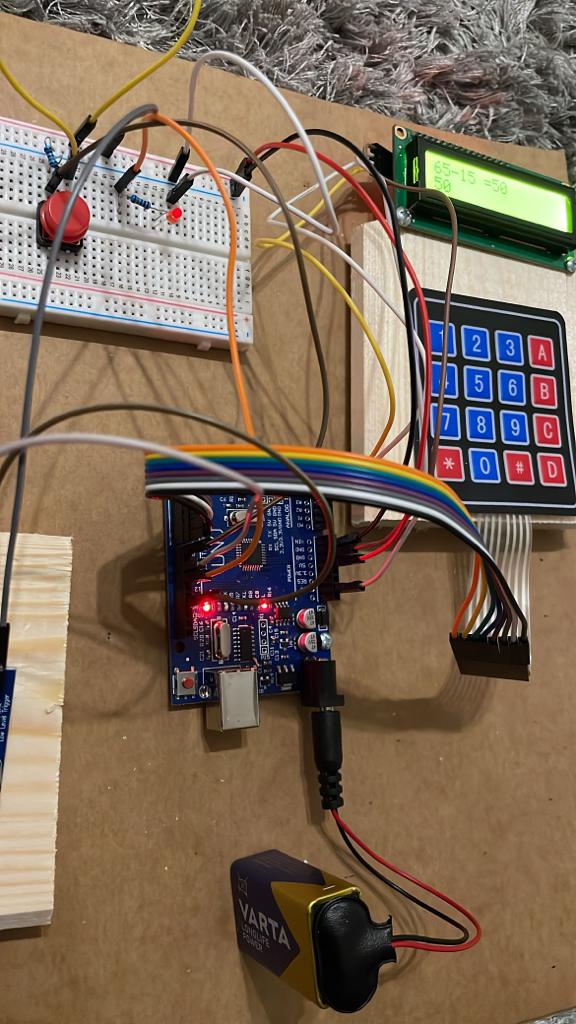
        }

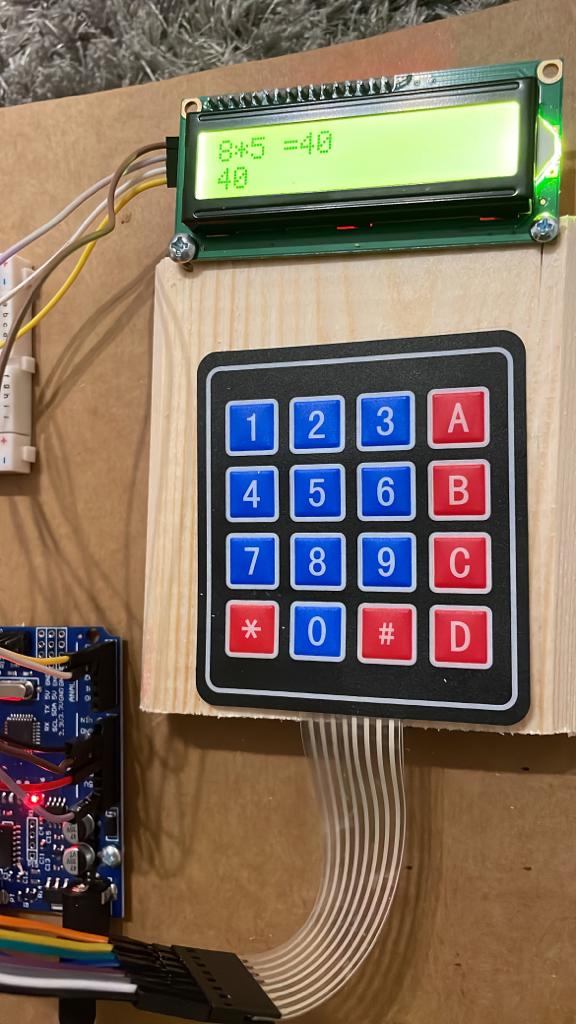
    }

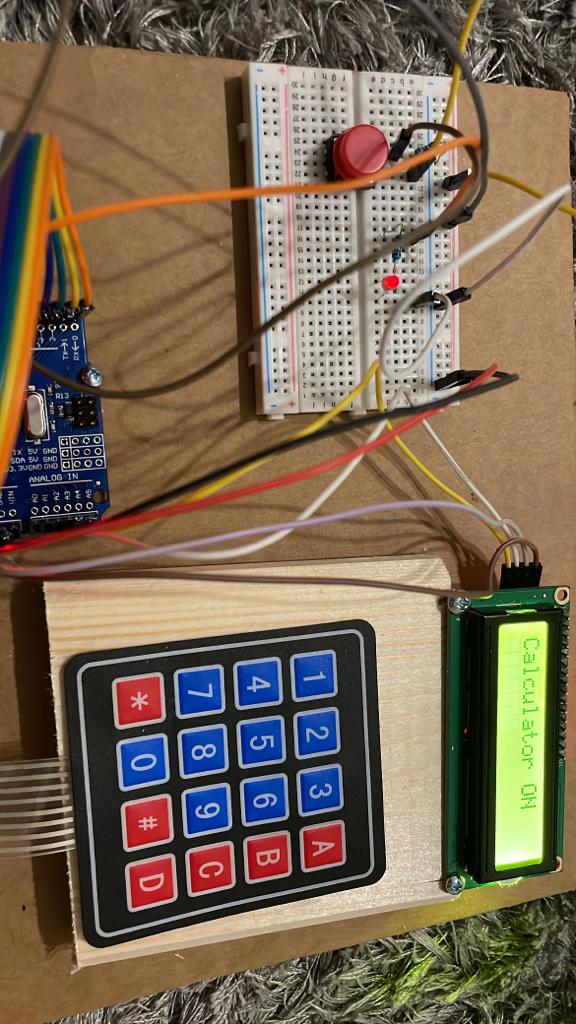
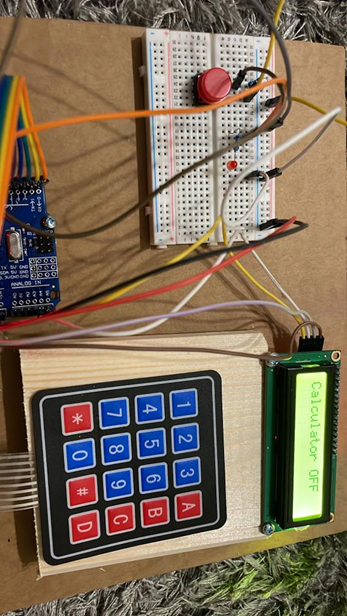
    lcd.setCursor(0, 1);

    lcd.print(Number); // Afișează rezultatul final pe ecran

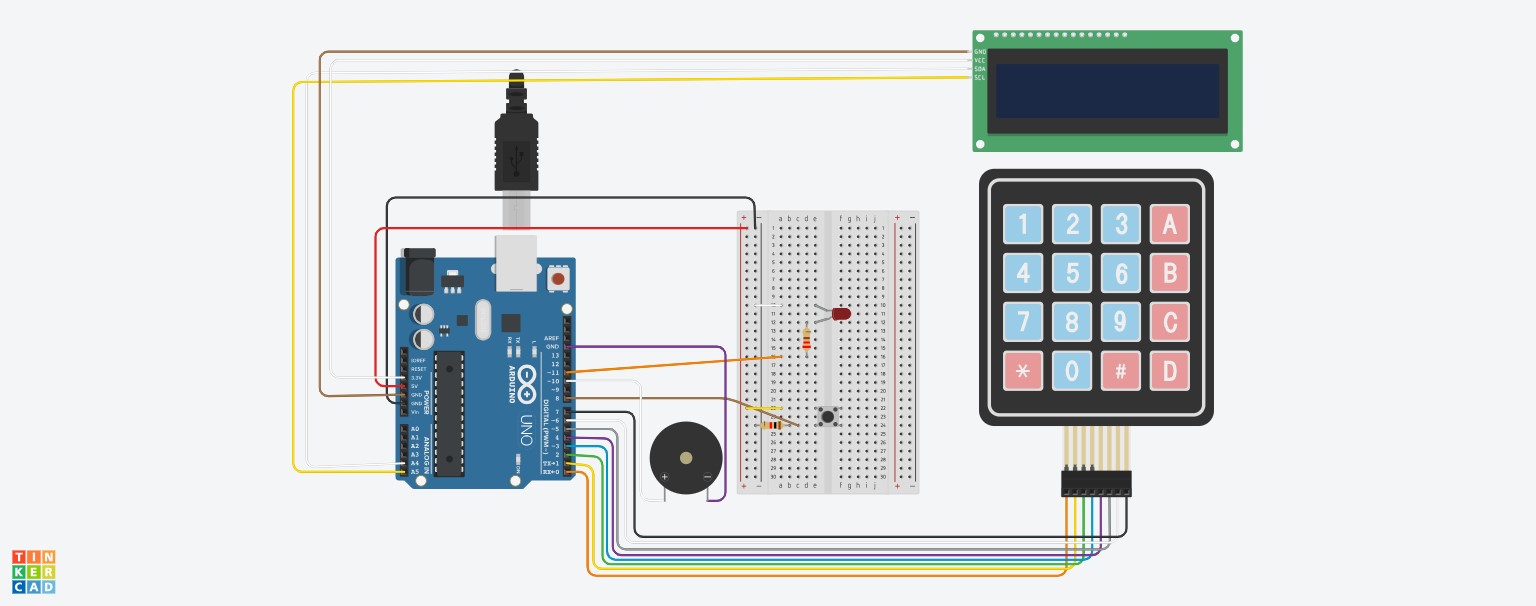
}

**IMAGINEA PROIECTULUI**



**SCHEMA MONTAJ**



**BIBLIOGRAFIE**

<https://www.youtube.com/watch?v=zl1o-t_17oQ&t=37s>

<https://www.youtube.com/watch?v=yBgMJssXqHY&t=240s>

<https://www.youtube.com/watch?v=FKekzzj5844&t=4s>

<https://www.youtube.com/watch?v=s_-nIgo71_w&t=304s>

<https://www.youtube.com/watch?v=wEbGhYjn4QI&t=2011s>

<https://www.youtube.com/watch?v=DbUrUQvcsMg&t=576s>

<https://www.youtube.com/watch?v=G2C6_Bp8cwk>

<https://www.youtube.com/watch?v=V_R0LAibhuc>

<https://www.youtube.com/watch?v=gj-H_agfd6U>